

University of Bahrain
 College of Information Technology
 Department of Computer Science
 Semester I, 2014-2015
 ITCS102/ITCS104/ITCS112 (Computer Programming II)

TEST 1

Date: 27/10/2014

Time: 11:00 - 12:15

STUDENT NAME	KEY SOLUTION
STUDENT ID #	
SECTION #	

QUESTION #	MARKS		COMMENTS
1-PART A	6		
1-PART B	12		
2-PART A	6		
2-PART B	11		
2-PART C	5		
TOTAL	40		

Question 1 [string Type]

PART A [6 Points]

Show the output of the following C++ program:

```
#include<iostream>
#include<string>
using namespace std;

int main( )
{
    string S1 = "Computer!", S2 = "Programming";

    int L = S1.size( );

    cout<<"Size = "<< L <<endl;

    cout<<"Found @ " <<S2.find('m')<<endl;

    string S3= S1 + "#is#" + S2 ;

    string S4 = S3.substr(9,7);

    cout<<"S3 = "<<S3<<endl;

    cout<<"S4 = "<<S4<<endl;

    return 0;
}
```

OUTPUT

// Each line 1.5 pt = Total = 6 pts

Size = 9

Found @ 6

S3 = Computer!#is#Programming

S4 = #is#Pro

Question 1 [Arrays]

PART B [12 Points]

Write a function named *insertZero* that takes as parameters: an array of integers (*list*), the number of elements in the array (*len*) passed by reference, and the maximum size of the array (*maxSize*).

The function should insert Zero after each positive number in the array that is followed by a negative number, and *len* should be incremented. The insertion should be done only if there is enough space in the array, i.e *len* is less than *maxSize*.

The Function prototype: `void insertZero(int list[], int &len, const int maxSize);`

For example, If *len*=8, *maxSize*=100 and the content of *list* before calling the function is:

0	1	2	3	4	5	6	7
11	-12	10	-10	4	3	-1	-5

Then, after calling the function: *len*=**11** and *list* should be as follows:

0	1	2	3	4	5	6	7	8	9	10
11	0	-12	10	0	-10	4	3	0	-1	-5

```
void insertZero(int list[ ], int &len, const int maxSize)
{
    // Total =12 pts
    for(int index=0; index<len-1 ; index++) // search 2 pts
        if(len==maxSize) // check size = 2pts
            break;
        else if (list[index]>0 && list[index+1]<0) // condition = 2pts
        {
            //shift to right = 3 pts
            for(int j=len; j>index; j--)
                list[j]=list[j-1];

            list[index+1]=0; // insert = 2 pts
            len++ ; // 1 pt
        }
}
```

Question 2 [Struct]

PART A [6 Points]

Define a struct named **laptop** with the following data members: ID, name, price and brand.

```
//Total = 6 pts
struct laptop{ // 1pt
    long ID; // 1 pt
    string name; // 1 pt
    float price; // 1 pt
    string brand; // 1 pt
}; // 1 pt
```

PART B [11 Points]

Write a function *binarySearch* which accepts an array of type **laptop**, array size, and an *IDKey*. The function should search in the array for the laptop with an ID equal to *IDkey*. If *IDkey* is found in the array, then the function should return its position (index) in the array, otherwise the function should return -1. The search should be done using the binary search algorithm, assuming that the array is sorted in ascending order using the laptop ID.

The function prototype is: `int binarySearch(laptop list[], int length, long IDKey);`

```
int binarySearch(laptop list[], int length, long IDKey )
{
    int first = 0; // 1 pt
    int last = length - 1; // 1pt
    int mid;

    while (first <= last) // 1 pt
    {
        mid = (first + last) / 2; // 1 pt
        if (list[mid].ID == IDKey) // 1 pt
            return mid; // 1 pt
        else if (list[mid].ID > IDKey) // 1 pt
            last = mid - 1; // 1 pt
        else // 1 pt
            first = mid + 1; // 1 pt
    }
    return -1; // 1 pt
}
```

Question 2 [Struct].. Continue

PART C [5 Points]

Write a main function to test the function *binarySearch* defined in PART (B). In the main function, declare an array of laptops of size 3. Then prompt the user to enter the laptops details (assuming that the user will enter the laptops sorted by ID in ascending order). Your program then should search for a laptop with ID =100123 using the function *binarySearch* defined in PART(B) and output an appropriate message.

```
int main( ){
    //Total = 5 pts
    //declaration = 1 pt
    laptop list[3];
    int len=3;

    // Read Data = 1.5 pts
    for(int i=0; i<3; i++)
    {
        cout<<"Enter ID, name, price, compnay name";
        cin>>list[i].ID>>list[i].name>>list[i].price>>list[i].brand;
    }

    // call function 1.5 pts
    int R = binarySearch(list,len,100123);

    // output data 1 pts
    if (R!=-1)
        cout<<"Found @ index = "<<R<<endl;
    else
        cout<<"Not Found..."<<endl;

    system("pause");
    return 0;
}
```

